

OUTPUT DEVICE FOR UNIVERSAL DATA OUTPUT

CROSS REFERENCE TO RELATED APPLICATION

[0001] This application claims benefit of Provisional Application No. 60/262,764 filed January 19, 2001, the entire disclosure of which is hereby incorporated by reference herein for all purposes.

TECHNICAL FIELD OF THE INVENTION

[0002] Present invention relates to providing content to an output device and, in particular, to providing universal output in which an information apparatus can pervasively output content to an output device without the need to install a dedicated device dependent driver or applications for each output device.

BACKGROUND OF THE INVENTION

[0003] The present invention relates to universal data output and, in particular, to providing a new data output method and a new raster image process for information apparatuses and output devices.

[0004] As described herein, information apparatuses refer generally to computing devices, which include both stationary computers and mobile computing devices (pervasive devices). Examples of such information apparatuses include, without limitation, desktop computers, laptop computers, networked computers, palmtop computers (hand-held computers), personal digital assistants (PDAs), Internet enabled mobile phones, smart phones, pagers, digital capturing devices (e.g., digital cameras and video cameras), Internet appliances, e-books, information pads, and digital or web pads. Output devices may include, without limitation, fax machines, printers, copiers, image and/or video display devices (e.g., televisions, monitors and projectors), and audio output devices.

[0005] For simplicity and convenience, hereafter, the following descriptions may refer to an output device as a printer and an output process as printing. However, it should be understood that the term printer and printing used in the discussion of present invention refer to one embodiment used as a specific example to simplify the description of the invention. The references to printer and printing used here are intended to be applied or extended to the larger scope and definition of output devices and

should not be construed as restricting the scope and practice of present invention.

[0006] Fueled by an ever-increasing bandwidth, processing power, wireless mobile devices, and wireless software applications, millions of users are or will be creating, downloading, and transmitting content and information using their pervasive or mobile computing devices. As a result, there is a need to allow users to conveniently output content and information from their pervasive computing devices to any output device. As an example, people need to directly and conveniently output from their pervasive information apparatus, without depending on synchronizing with a stationary computer (e.g., desktop personal computer) for printing.

[0007] To illustrate, a mobile worker at an airport receiving e-mail in his hand-held computer may want to walk up to a nearby printer or fax machine to have his e-mail printed. In addition, the mobile worker may also want to print a copy of his to-do list, appointment book, business card, and his flight schedule from his mobile device. As another example, a user visiting an e-commerce site using his mobile device may want to print out transaction confirmation. In still another example, a user who takes a picture with a digital camera may want to easily print it out to a nearby printer. In any of the above cases, the mobile user may want to simply walk up to a printer and conveniently print a file (word processing document, PDF, HTML etc) that is stored on the mobile device or downloaded from a network (e.g., Internet, corporate network).

[0008] Conventionally, an output device (e.g., a printer) is connected to an information apparatus via a wired connection such as a cable line. A wireless connection is also possible by using, for example, radio communication or infrared communication. Regardless of wired or wireless connection, a user must first install in the information apparatus an output device driver (e.g., printer driver in the case the output device is a printer) corresponding to a particular output device model and make. Using a device-dependent or specific driver, the information apparatus may process output content or digital document into a specific output device's input requirements (e.g., printer input requirements). The output device's input requirements correspond to the type of input that the output device (e.g., a printer) understands. For example, a printer's input requirement may include printer specific input format (e.g., one or more of an image,

graphics or text format or language). Therefore, an output data (or print data in the case the output device is a printer) herein refers to data that is acceptable for input to an associated output device. Examples of input requirements may include, without limitation, audio format, video format, file format, data format, encoding, language (e.g., page description language, markup language etc), instructions, protocols or data that can be understood or used by a particular output device make and model.

[0009] Input requirements may be based on proprietary or published standards or a combination of the two. An output device's input requirements are, therefore, in general, device dependent. Different output device models may have their own input requirements specified, designed or adopted by the output device manufacturer (e.g., the printer manufacturer) according to a specification for optimal operation. Consequently, different output devices usually require use of specific output device drivers (e.g., printer drivers) for accurate output (e.g., printing). Sometimes, instead of using a device driver (e.g., printer driver), the device driving feature may be included as part of an application software.

[0010] Installation of a device driver (e.g., printer driver) or application may be accomplished by, for example, manual installation using a CD or floppy disk supplied by the printer manufacturer. Or alternatively, a user may be able to download a particular driver or application from a network. For a home or office user, this installation process may take anywhere from several minutes to several hours depending on the type of driver and user's sophistication level with computing devices and networks. Even with plug-and-play driver installation, the user is still required to execute a multi-step process for each printer or output device.

[0011] This installation and configuration process adds a degree of complexity and work to end-users who may otherwise spend their time doing other productive or enjoyable work. Moreover, many unsophisticated users may be discouraged from adding new peripherals (e.g., printers, scanners, etc.) to their home computers or networks to avoid the inconvenience of installation and configuration. It is therefore desirable that an information apparatus can output to more than one output device without the inconvenience of installing multiple dedicated device dependent drivers.

[0012] In addition, conventional output or printing methods may pose significantly higher challenges and difficulties for mobile device users than for home and office users. The requirement for pre-installation of a device-dependent driver diminishes the benefit and concept of mobile (pervasive) computing and output. For example, a mobile user may want to print or output e-mail, PowerPoint® presentation documents, web pages, or other documents at an airport, gas station, convenience store, kiosk, hotel, conference room, office, home, etc. It is highly unlikely that the user would find at any of these locations a printer of the same make and model as is at the user's base station. As a consequence, under the conventional printing method, the user would have to install and configure a printer driver each time at each such remote location before printing. It is usually not a viable option given the hundreds, or even thousands of printer models in use, and the limited storage, memory space, and processing power of the information apparatus.

[0013] Moreover, the user may not want to be bothered with looking for a driver or downloading it and installing it just to print out or display one page of email at the airport. This is certainly an undesirable and discouraging process to promote pervasive or mobile computing. Therefore, a more convenient printing method is needed in support of the pervasive computing paradigm where a user can simply walk up to an output device (e.g., printer or display device) and easily output a digital document without having to install or pre-install a particular output device driver (e.g., printer driver).

[0014] Another challenge for mobile users is that many mobile information apparatuses have limited memory space, processing capacity and power. These limitations are more apparent for small and low-cost mobile devices including, for example, PDAs, mobile phones, screen phones, pagers, e-books, Internet Pads, Internet appliances etc. Limited memory space poses difficulties in installing and running large or complex printer or device drivers, not to mention multiple drivers for a variety of printers and output devices. Slow processing speed and limited power supply create difficulties driving an output device. For example, processing or converting a digital document into output data by a small mobile information apparatus may be so slow that it is not suitable for productive output. Intensive processing may also drain or consume power or battery resources. Therefore, a method is needed so that a small mobile device, with limited processing

capabilities, can still reasonably output content to various output devices.

[0015] To output or render content (e.g. digital document) to an output device, a raster image processing (RIP) operation on the content is usually required. RIP operation can be computationally intensive and may include (1) a rasterization operation, (2) a color space conversion, and (3) a halftoning operation. RIP may also include other operations such as scaling, segmentation, color matching, color correction, GCR (Grey component replacement), Black generation, image enhancement compression/decompression, encoding/decoding, encryption/decryption GCR, image enhancement among others.

[0016] Rasterization operation in RIP involves converting objects and descriptions (e.g. graphics, text etc) included in the content into an image form suitable for output. Rasterization may include additional operations such as scaling and interpolation operations for matching a specific output size and resolution. Color space conversion in RIP includes converting an input color space description into a suitable color space required for rendering at an output device (e.g. RGB to CMYK conversion). Digital halftoning is an imaging technique for rendering continuous tone images using fewer luminance and chrominance levels. Halftoning operations such as error diffusion can be computationally intensive and are included when the output device's bit depth (e.g. bits per pixel) is smaller than the input raster image bit depth.

[0017] Conventionally, RIP operations are included either in an information apparatus, or as part of an output device or output system (e.g. in a printer controller). Fig. 1A illustrates a flow diagram of a conventional data output method 102 in which RIP 110 is implemented in the information apparatus. Output devices that do not include a printer controller to perform complex RIP operations, such as a lower-cost, lower speed inkjet printer, normally employ data output method 102. In data output method 102, an information apparatus obtains content (e.g. a digital document) in step 100 for rendering or output at an output device. The information apparatus may include an application (e.g. device driver), which implements RIP operation 110. The information apparatus generates an output data in step 120 and transmits the output data to the output device in step 130 for rendering. The output data relating to the content is in an acceptable form (e.g. in an appropriate output size and resolution) to the output engine (e.g. display engine, printer

engine etc.) included in the output device. The output data in a conventional output method 102 is usually device dependent.

[0018] One drawback for the data output method 102 of Fig. 1A is that the information apparatus performs most if not the entire raster image processing operations 110 required for output. The RIP operations may require intensive computation. Many information apparatus such as mobile information device might have insufficient computing power and/or memory to carry out at an acceptable speed the RIP operations 110 required in an output process.

[0019] Another drawback for the conventional data output method 102 of Fig. 1A is that the generated output data is device dependent and therefore is typically not very portable to other output devices. As a result, the information apparatus may need to install multiple applications or device drivers for multiple output devices, which may further complicate its feasibility for use in information apparatuses with limited memory, storage and processing power.

[0020] Fig. 1B illustrates a flow diagram of another conventional data output method 104 in which the RIP is implemented in an output device. An example of an output device that implements process 104 is a high-speed laser printer which includes a printer controller for performing RIP operations and an output engine (e.g. printer engine) for rendering content. Printer controller may be internally installed or externally connected to an output device (printer in this example). In data output method 104, an information apparatus obtains content for output in step 100 and generates in step 160 an output data or print data for transmitting to the output device in step 170. Print data includes information related to the content and is usually encoded in a page description language (PDL) such as PostScript and PCL etc. In step 180, the printer receives the output data or print data (in a PDL). In step 190, a printer controller included in the printer interprets the PDL, performs RIP operations, and generates a printer-engine print data that is in a form acceptable to the printer engine (e.g. a raster image in an appropriate output size, bit depth, color space and resolution). In step 150 the printer engine renders the content with the printer-engine print data.

[0021] It will be understood that a reference to print data or output data including a language, such as PDL, should be interpreted as meaning that the print data or output data is encoded using that language. Correspondingly, a reference to a data output process generating a language, such as PDL, should be

interpreted as meaning that the data output process encodes data using that language.

[0022] There are many drawbacks in the conventional data output method 104 shown in Fig 1B. These drawbacks are especially apparent for mobile computing devices with limited processing power and memory. One such drawback is that the output data or print data, which include a page description language (PDL) such as PostScript or PCL, can be very complex. Generating complex PDL may increase memory and processing requirements for an information apparatus. Furthermore, interpreting, decoding and then raster image processing complex PDL can increase computation, decrease printing speed, and increase the cost of the output device or its printer controller.

[0023] Another drawback is that the output data that includes PDL can create a very large file size that would increase memory and storage requirements for the information apparatus, the output device and/or the printer controller etc. Large file size may also increase the bandwidth required in the communication link between the information apparatus and the output device.

[0024] Finally, to rasterize text in an output device, a printer controller may need to include multiple fonts. When a special font or international characters is not included or missing in the printer controller, the rendering or output can potentially become inaccurate or inconsistent.

SUMMARY OF THE INVENTION

[0025] Accordingly, this invention provides a convenient universal data output method in which an information apparatus and an output device or system share the raster image processing operations. Moreover, the new data output method eliminates the need to install a plurality of device-dependent dedicated drivers or applications in the information apparatus in order to output to a plurality of output devices.

[0026] In accordance with present invention, an electronic system and method of pervasive and universal output allow an information apparatus to output content conveniently to virtually any output device. The information apparatus may be equipped with a central processing unit, input/output control unit, storage unit, memory unit, and wired or wireless communication unit or adapters. The information apparatus preferably includes a client application that may be implemented as a software application, a helper application, or a device driver (a printer driver in case of a

printer). The client application may include management and control capabilities with hardware and software components including, for example, one or more communication chipsets residing in its host information apparatus.

[0027] The client application in the information apparatus may be capable of communicating with, managing and synchronizing data or software components with an output device equipped with an output controller of present invention.

[0028] Rendering content in an output device refers to printing an image of the content onto an substrate in the case of a printing device; displaying an image of the content in the case of a displaying device; playing an audio representation of the content in a voice or sound output device or system.

[0029] An output controller may be a circuit board, card or software components residing in an output device. Alternatively, the output controller may be connected externally to an output device as an external component or "box." The output controller may be implemented with one or more combinations of embedded processor, software, firmware, ASIC, DSP, FPGA, system on a chip, special chipsets, among others. In another embodiment, the functionality of the output controller may be provided by application software running on a PC, workstation or server connected externally to an output device.

[0030] In conventional data output method 102 as described with reference to Fig. 1A, an information apparatus transmits output data to an output device for rendering. Output data corresponds to content intended for output and is mostly raster image processed (RIPed) and therefore is device dependent because raster image processing is a typical device dependent operation. Output data may be encoded or compressed with one or more compression or encoding techniques. In present invention, an information apparatus generates an intermediate output data for transmitting to an output device. The intermediate output data includes a rasterized image corresponding to the content; however, device dependent image processing operations of a RIP (e.g. color matching and halftoning) have not been performed. As a result, an intermediate output data is more device independent and is more portable than the output data generated by output method with reference to Fig 1A.

[0031] In one implementation of this invention, the intermediate output data includes MRC (Mixed raster content) format, encoding and compression techniques, which further provides

improved image quality and compression ratio compared to conventional image encoding and compression techniques.

[0032] In an example of raster image process and data output method of the present invention, a client application such as a printer driver is included in an information apparatus and performs part of raster image processing operation such as rasterization on the content. The information apparatus generates an intermediate output data that includes an output image corresponding to the content and sends the intermediate output data to an output device or an output system for rendering. An output controller application or component included in the output device or output system implements the remaining part of the raster image processing operations such as digital halftoning, color correction among others.

[0033] Like conventional raster image processing methods, this invention provides a more balanced distribution of the raster image processing computational load between the Information apparatus and the output device or the output system. Computational intensive image processing operations such as digital halftoning and color space conversions can be implemented in the output device or output system. Consequently, this new raster image processing method reduces the processing and memory requirements for the information apparatus when compared to conventional data output methods described with reference to Fig 1A in which the entire raster image process is implemented in the information apparatus. Additionally, in this invention, a client application or device driver included in the information apparatus, which performs part of the raster image processing operation, can have a smaller size compared to a conventional output application included in the information apparatus, which performs raster image processing operation.

[0034] In another implementation, the present invention provides an information apparatus with output capability that is more universally accepted by a plurality of output devices. The information apparatus, which includes a client application, generates an intermediate output data that may include device independent attributes. An output controller includes components to interpret and process the intermediate output data. The information apparatus can output content to different output devices or output systems that include the output controller even when those output devices are of different brand, make, model and with different output engine and input data requirements. Unlike conventional output methods, a user does not need to preinstall in

the information apparatus multiple dedicated device dependent drivers or applications for each output device.

[0035] The combination of a smaller-sized client application, a reduced computational requirement in the information apparatus, and a more universal data output method acceptable for rendering at a plurality of output devices enable mobile devices with less memory space and processing capabilities to implement data output functions which otherwise would be difficult to implement with conventional output methods.

[0036] In addition, this invention can reduce the cost of an output device or an output system compared to conventional output methods 104 that include a page description language (PDL) printer controller. In the present invention, an information apparatus generates and sends an intermediate output data to an output device or system. The intermediate output data in one preferred embodiment includes a rasterized output image corresponding to the content intended for output. An output controller included in an output device or an output system decodes and processes the intermediate output data for output, without performing complex interpretation and rasterization compared to conventional methods described in process 104. In comparison, the conventional data output process 104 generates complex PDL and sends this PDL from an information apparatus to an output device that includes a printer controller (e.g. a PostScript controller or a PCL5 controller among others). Interpretation and raster image processing of a PDL have much higher computational requirements compared to decoding and processing the intermediate output data of this invention that include rasterized output image or images. Implementing a conventional printer controller with, for example, PDL increases component cost (e.g. memories, storages, ICs, software and processors etc.) when compared to using the output controller included in the data output method of this present invention.

[0037] Furthermore, an output data that includes PDL can create a large file size compared to an intermediate output data that includes rasterized output image. The data output method for this invention comparatively transmits a smaller output data from an information apparatus to an output device. Smaller output data size can speed up transmission, lower communication bandwidth, and reduce memory requirements. Finally, this invention can provide a convenient method to render content at an output device with or without connection to a static network. In conventional network printing, both information apparatus and output device must be

connected to a static network. In this invention, through local communication and synchronization between an information apparatus and an output device, installation of hardware and software to maintain static network connectivity may not be necessary to enable the rendering of content to an output device.

[0038] According to the several aspects of the present invention there is provided the subject matter defined in the appended independent claims.

[0039] Additional objects and advantages of the present invention will be apparent from the detailed description of the preferred embodiment thereof, which proceeds with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0040] Fig. 1A is a flow diagram of a conventional data output method and its corresponding raster image process in accordance with prior art.

[0041] Fig. 1B is a flow diagram of a second conventional data output method and its corresponding raster image process for an output device that includes a conventional printer controller in accordance with prior art.

[0042] Figs. 2A and 2B are block diagrams illustrating components of an operating environment that can implement the process and apparatus of the present invention.

[0043] Fig. 3A is a schematic block diagram illustrating hardware/software components of an information apparatus implementation in accordance with present invention. The information apparatus includes an operating system.

[0044] Fig. 3A is a second schematic block diagram illustrating hardware/software components of an information apparatus implementation in accordance with present invention.

[0045] Fig. 4A is a block diagram of a conventional printing system or printer with a conventional printer controller.

[0046] Fig. 4B is a block diagram of a second conventional output system or output device.

[0047] Fig. 5A is a schematic block diagram of a printing system or printer with a conventional printer controller and an output controller in accordance with present invention.

[0048] Fig. 5B is a schematic block diagram of a second output system or output device that includes an output controller in accordance with present invention.

[0049] Fig. 6A is a schematic block diagram illustrating hardware/software components of an output controller in accordance with present invention. The output controller includes an operating system.

[0050] Fig. 6B is a second schematic block diagram illustrating hardware/software components of an output controller in accordance with present invention. The output controller does not include an operating system.

[0051] Fig. 6C is a third schematic block diagram illustrating hardware/software components of an output controller in accordance with present invention. The output controller combines the functionality of a printer controller and an output controller of present invention.

[0052] Figs. 7A-7F illustrate various configurations and implementations of output controller with respect to an output device such as a printer.

[0053] Fig. 8A is a block diagram illustrating an exemplary implementation of hardware/software components of wireless communication unit.

[0054] Fig. 8B is block diagram illustrating a second exemplary implementation of hardware/software components of wireless communication unit.

[0055] Fig. 9 is a flow diagram of a universal data output method and its corresponding raster imaging process of the present invention.

[0056] Fig. 10 is a block diagram of a universal data output method of the present invention with respect to the components, system and apparatus described with reference to Fig 2.

[0057] Fig. 11 is a flow diagram illustrating one way of implementing a discovery process optionally included in the output process of Fig. 10.

[0058] Fig. 12A and 12B are flow diagrams of exemplary client application process included in the output process of Fig. 10.

[0059] Fig. 13A and 13B are flow diagrams of exemplary output device or output system process included in the output process of Fig. 10.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[0060] Sets forth below are definitions of terms that are used in describing implementations of the present invention. These definitions are provided to facilitate understanding and illustration of implementations of the present invention and should

in no way be construed as limiting the scope of the invention to a particular example, class, or category.

Output device profile (or object)

[0061] An output device profile (or object) includes software and data entity, which encapsulates within itself both data and attributes describing an output device and instructions for operating that data and attributes. An output device profile may reside in different hardware environments or platforms or applications, and may be transported in the form of a file, a message, a software object or component among other forms and techniques. For simplicity of discussion, a profile or object may also include, for example, the concept of software components that may have varying granularity and can consist of one class, a composite of classes, or an entire application.

[0062] The term profile or object used herein is not limited to software or data as its media. Any entity containing information, descriptions, attributes, data, instructions etc. in any computer-readable form or medium such as hardware, software, files based on or including voice, text, graphics, image, or video information, electronic signals in analog or digital form, etc., are all valid forms of profile and object definition.

[0063] A profile or object may also contain in one of its fields or attributes a reference or pointer to another profile or object, or a reference or pointer to data and or content. A reference to a profile or object may include one or more, or a combination of pointers, identifiers, names, paths, addresses or any descriptions relating to a location where an object, profile, data, or content can be found.

[0064] An output device profile may contain one or more attributes that may identify and describe, for example, the capabilities and functionalities of a particular output device such as a printer. An output device profile may be stored in the memory component of an output device, an information apparatus or in a network node. A network node includes any device, server or storage location that is connected to the network. As described below in greater detail, an information apparatus requesting output service may communicate with an output device. During such local service negotiation, at least a partial output device profile may be uploaded to the information apparatus from the output device. By obtaining the output device profile (or printer profile in the case of a printer), the information apparatus may learn about the

capability, compatibility, identification, and service provided by the output device.

[0065] As an example, an output device profile may contain one or more of the following fields and or attribute descriptions. Each of following fields may be optional, and furthermore, each of the following fields or attributes may or may not exist in a particular implementation (e.g., may be empty or NULL):

[0066] Identification of an output device (e.g., brand, model, registration, IP address etc.)

- Services and feature sets provided by an output device (e.g., color or grayscale output, laser or inkjet, duplex, output quality, price per page, quality of service, etc.)

- Type of input languages, formats, output data and/or input requirements (e.g., PostScript, PCL, XML, RTL, etc.) supported by an output device.

- Device specific or dependent parameters and information (e.g., communication protocols, color space, color management methods and rendering intents, resolution, halftoning methods, dpi (dots-per-inch), bit depth, page size, printing speed, number of independent colors channels or ink etc.)

- Data and tables needed for image processing such as color table, halftone table, scale factor, encoding/decoding parameters and methods, compression and decompression parameters and method etc.

- Another profile which contain parameters and information about the output device and its service (e.g. color profiles, halftoning profiles, communication profiles, rasterization profiles, quality of service etc.).

- Payment information on a plurality of services provided by an output device.

- Information or security requirements and type of authentication an output device supports.

- Date and version of the output device profile, history of its modification and updates.

- Software components containing algorithms or instructions or data, which may be uploaded to run in an information apparatus. For example, a graphical user interface (GUI) software component may be uploaded to an information apparatus. The software component may be incorporated into or launched in the information apparatus by a client application of present invention to capture a user's preferences (e.g., print quality, page layout, number of copies, number of cards per page, etc.). In another example,

software components may include methods, instructions or executables for compression/decompression, encoding/decoding, color matching or correction, segmentation, scaling, halftoning, encryption/decryption among others.

- Pointer or reference to one or more output device parameters, including one or more of the above described output device profile or object fields and or attribute descriptions. For example, a more up-to-date or original version of output device parameters may sometimes be stored in a network node (any device, server or storage location that is connected to the network), or within the information apparatus where it can be obtained by the client application. An output device profile may include pointer or pointers to these output device parameters.

Content (or data content, digital content, output content)

[0067] Content (or data content, digital content, output content) is the data intended for output, which may include texts, graphics, images, forms, videos, audio among other content types. Content may include the data itself or a reference to that data. Content may be in any format, language, encoding or combination, and it can be in a format, language or encoding that is partially or totally proprietary. A digital document is an example of content that may include attributes and fields that describe the digital document itself and or reference or references to the digital document or documents. Examples of a digital document may be any one or combination of file types: HTML, VHTML, PostScript, PCL, XML, PDF, MS Word, PowerPoint, JPEG, MPEG, GIF, PNG, WML, VWML, CHTML, HDML, ASCII, 2-byte international coded characters, etc. Content may be used interchangeably with the term data content, output content or digital content in the descriptions of present invention.

Intermediate output data

[0068] Output data (or print data in case of a printer) is the electronic data sent from an information apparatus to an output device. Output data is related to the content intended for output and may be encoded in a variety of formats and languages (e.g. postscript, PCL, XML), which may include compressed or encrypted data. Some output device manufacturers may also include in the output data (or print data) a combination of proprietary or non-proprietary languages, formats, encoding, compression, encryption etc.

[0069] Intermediate output data is the output data of the present invention, and it includes the broader definition of an output file or data generated by an information apparatus, or a client application or device driver included in the information apparatus. An intermediate output data may contain text, vector graphics, images, video, audio, symbols, forms or combination and can be encoded with one or more of a page description language, a markup language, a graphics format, an imaging format, a metafile among others. An intermediate output data may also contain instructions (e.g. output preferences) and descriptions (e.g. data layout) among others. Part or all of an intermediate output data may be compressed, encrypted or tagged.

[0070] In a preferred embodiment of this invention, intermediate output data contains rasterized image data. For example, vector graphics and text information or objects that are not in image form included in content can be rasterized or conformed into image data in an information apparatus and included in an intermediate output data. Device dependent image processing operations of a RIP such as digital halftoning and color space conversions can be implemented at an output device or an output system.

[0071] The intermediate output data can be device dependent or device independent. In one implementation, the rasterized output image is device dependent if the rasterization parameters used, such as resolution, scale factor, bit depth, output size and or color space are device dependent. In another implementation of this invention, the rasterized image may be device independent if the rasterization parameters used are device independent. Rasterization parameter can become device independent when those parameters include a set of predetermined or predefined rasterization parameters based on a standard or a specification. With predefined or device independent rasterization parameters, a client application of present invention can rasterize at least a portion of the content and generate a device independent image or images included in the intermediate output data. By doing so, the intermediate output data may become device independent and therefore, become universally acceptable with output devices that have been pre-configured to accept the intermediate output data.

[0072] One advantage of rasterizing or converting text and graphics information into image data at the information apparatus is that the output device or printer controller no longer needs to perform complex rasterization operation nor do they need to include

multiple fonts. Therefore, employing the intermediate output data and the data output method described herein could potentially reduce the cost and complexity of an output controller, printer controller and or output device.

[0073] One form of image data encoding is known as mixed raster content, or MRC. Typically, an image stored in MRC includes more than one image or bitmap layers. In MRC, an image can be segmented in different layers based on segmentation criteria such as background and foreground, luminance and chrominance among others. For example, an MRC may include three layers with a background layer, a foreground layer and a toggle or selector layer. The three layers are coextensive and may include different resolution, encoding and compression. The foreground and background layers may each contain additional layers, depending on the manner in which the respective part of the image is segmented based on the segmentation criteria, component or channels of a color model, image encoding representation (HLS, RGB, CMYK, YCC, LAB etc) among others. The toggle layer may designate, for each point, whether the foreground or background layer is effective. Each layer in a MRC can have different bit depths, resolution, color space, which allow, for example, the foreground layer to be compressed differently from the background layer. The MRC form of image data has previously been used to minimize storage requirements. Further, an MRC format has been proposed for use in color image fax transmission.

[0074] In one embodiment of present invention, the intermediate output data includes one or more rasterized output images that employ MRC format, encoding and or related compression method. In this implementation, different layers in the output image can have different resolutions and may include different compression techniques. Different information such as chrominance and luminance and or foreground and background information in the original content (e.g. digital document) can be segmented and compressed with different compression or encoding techniques. Segmented elements or object information in the original content can also be stored in different image layers and with different resolution. Therefore, with MRC, there is opportunity to reduce output data file size, retain greater image information, increase compression ratio, and improve image quality when compared to other conventional image encoding and compression techniques. Implementations of rasterization, raster image processing and

intermediate output data that include MRC encoding in the present invention are described in more detail below.

Rasterization

[0075] Rasterization is an operation by which graphics and text in a digital document are converted to image data. For image data included in the digital document, rasterization may include scaling and interpolation. The rasterization operation is characterized by rasterization parameters including, among others bit depth and resolution. A given rasterization operation may be characterized by several more rasterization parameters, including output size, color space, color channels etc. Values of one or more of the rasterization parameters employed in a rasterization operation may be specified by default; values of one or more of the rasterization parameters may be supplied to the information apparatus as components of a rasterization vector. In a given application, the rasterization vector may specify a value of only one rasterization parameter, default values being employed for other rasterization parameters used in the rasterization operation. In another application the rasterization vector may specify values of more than one, but less than all, rasterization parameters, default values being employed for at least one other rasterization parameter used in the rasterization operation. And in yet another application the rasterization vector may specify values of all the rasterization parameters used in the rasterization operation.

[0076] Fig. 2A and 2B are block diagrams illustrating components of an operating environment that can implement the process and apparatus of present invention. Fig 2A shows an electronic system which includes an information apparatus 200 and an output device 220. The output device 220 includes an output controller 230. Fig. 2B illustrates a second implementation of an electronic system that includes an information apparatus 200 and an output system 250. The output system 250 includes an output device 220 and an output controller 230 which may be externally connected to, or otherwise associated with, the output device 220 in the output system 250.

[0077] Information apparatus 200 is a computing device with processing capability. In one embodiment, information apparatus 200 may be a mobile computing device such as palmtop computer, handheld device, laptop computer, personal digital assistant (PDA), smart phone, screen phone, e-book, Internet pad, communication pad, Internet appliance, pager, digital camera, etc. It is possible

that information apparatus 200 may also include a static computing device such as a desktop computer, workstation, server, etc.

[0078] Fig. 3A and 3B are block diagrams illustrating examples of hardware/software components included in an information apparatus 200 of present invention.

[0079] Information apparatus 200 may contain components such as a processing unit 380, a memory unit 370, an optional storage unit 360 and an input/output control unit (e.g. communication manager 330). Information apparatus 200 may include an interface (not shown) for interaction with users. The interface may be implemented with software or hardware or a combination. Examples of such interfaces include, without limitation, one or more of a mouse, a keyboard, a touch-sensitive or non-touch-sensitive screen, push buttons, soft keys, a stylus, a speaker, a microphone, etc.

[0080] Information apparatus 200 typically contains one or more network communication unit 350 that interfaces with other electronic devices such as network node (not shown), output device 220, and output system 230. The network communication unit may be implemented with hardware (e.g., silicon chipsets, antenna), software (e.g., protocol stacks, applications) or a combination.

[0081] In one embodiment of the present invention, communication interface 240 between information apparatus 200 and output device 220 or output system 250 is a wireless communication interface such as a short-range radio interface including those implemented according to the Bluetooth or IEEE 802.11 standard. The communication interface may also be realized by other standards and/or means of wireless communication that may include radio, infrared, cellular, ultrasonic, hydrophonic among others for accessing one or more network node and/or devices. Wired line connections such as serial or parallel interface, USB interface and fire wire (IEEE 1394) interface, among others, may also be included. Connection to a local network such as an Ethernet or a token Ring network, among others, may also be implemented in the present invention for local communication between information apparatus 200 and output device 220. Examples of hardware/software components of communication units 350 that may be used to implement wireless interface between the information apparatus 200 and the output device 220 are described in more detail with reference to Figs. 8A and 8B below.

[0082] For simplicity, Fig. 3 illustrates one implementation where an information apparatus 200 includes one communication unit 350. However, it should be noted that an information apparatus 200

[0083] Information apparatus 200 may be a dedicated device (e.g., email terminal, web terminal, digital camera, e-book, web pads, Internet appliances etc.) with functionalities that are pre-configured by manufacturers. Alternatively, information apparatus 200 may allow users to install additional hardware components and/or application software 205 to expand its functionality.

[0085] As illustrated previously with reference to Fig. 1, there are many difficulties in providing output capability to an information apparatus 200 that has limited memory and processing capability. To address these difficulties, information apparatus 200 includes a client application 210 that helps provide the universal data output capability of the present invention. Client

[0085] As illustrated previously with reference to Fig. 1, there are many difficulties in providing output capability to an information apparatus 200 that has limited memory and processing capability. To address these difficulties, information apparatus 200 includes a client application 210 that helps provide the universal data output capability of the present invention. Client

application 210 may include software and data that can be executed by the processing unit 380 of information apparatus 200. Client application 210 may be implemented as a stand-alone software application or as a part of or feature of another software application, or in the form of a device driver, which may be invoked, shared and used by other application software 205 in the information apparatus 200. Client application 210 may also include components to invoke other applications 205 (e.g., a document browsing application, editing application, data and/or image acquisition application, a communication manager, a output manager etc.) to provide certain feature sets, as described below. Fig 3 illustrates a configuration where the client application 210 is a separate application from the other application 205 such as the case when the client application is a device driver; however, it should be noted that the client application 210 can be combined or being part of the other application not shown in Fig 3. Client application 210 may be variously implemented in an information apparatus 200 and may run on different operating systems or platforms. The client application 210 may also run in an environment with no operating system. For example, Fig. 3A illustrates an implementation where the information apparatus 200A includes an operating system 340A; while Fig. 3B illustrates an implementation where the information apparatus 200B does not include an operating system.

[0086] Client application 210 includes a rasterization component 310 to conform content into one or more raster output images according to one or more rasterization parameters; an intermediate output data generator component 320 that generates and/or encodes intermediate output data that includes the one or more output images; and a communications manager 330 that manages the communication and interaction with an output device 220 or system 250 or output controller 230. Communications manager can be implemented as part of the client application 210 (shown in Fig. 3) or as a separate application (not shown). Components in a client application can be implemented in software, hardware or combination. As an example, client application 210 may include or utilize one or more of the following:

- Components or operations to obtain content (e.g. digital document) for output. The client application 210 may obtain a digital document from other applications 205 (e.g. document browsing application, content creation and editing application,

etc.), or the client application 210 may provide its own capability for user to browse, edit and or select a digital document.

- Components or operations to rasterize content that includes text, graphics and images among others objects or elements into one or more raster images according to a set of rasterization parameters such as scale factor, output size, bit depth, color space and resolution. The rasterization parameters may be obtained in various ways, for example, from an output device profile uploaded from an output device 220, or stored locally in information apparatus 200, or manually inputted by a user. Alternatively, rasterization parameters may be based on a predefined standard or specification stored in the information apparatus 200 as a set of defaults, or hard-coded in the client application 210, or calculated by the client application 210 after communicating with an output device 220, output controller 230, and/or a user.

- Components or operations to generate intermediate output data that includes at least one rasterized output image corresponding to the content (e.g. digital document). This process may further include one or combination of compression, encoding, encryption and color correction among others. The intermediate output data may include, for example, images, instructions, documents and or format descriptions, color profiles among others.

- Components or operations to transmit the intermediate output data to an output device 220 or system 250 through wired or wireless communication link 240.

[0087] The client application 210 may also optionally include or utilize one or more of the following components or operations:

- Components or operations to communicate with one or more output devices 220 to upload an output device profile.

- Components or operations to communicate directly or indirectly (such as through an operating system or component or object model, messages, file transfer etc.) with other applications 205 residing in the same information apparatus 200 to obtain objects, data, and or content needed, or related to the pervasive output process of present invention (e.g. obtain a digital document for printing).

- Components or operations to manage and utilize directly or indirectly functionalities provided by hardware components (e.g. communication unit 350) residing in its host information apparatus 200.

- Components or operations to provide a graphical user

interface (GUI) in host information apparatus to interact with user.

- Components or operations to obtain user preferences. For example, a user may directly input his or her preferences through a GUI. A set of default values may also be employed. Default values may be pre-set or may be obtained by information apparatus 200 as result of communicating and negotiating with an output device 220 or output controller 230.

[0088] The above functionalities and process of client application 210 of present invention are described in further detail in the client application process with reference to FIG 12.

[0089] Output device 220 is an electronic system capable of outputting digital content regardless of whether the output medium is substrate (e.g., paper), display, projection, or sound. A typical example of output device 220 is a printer, which outputs digital documents containing text, graphics, image or any combination onto a substrate. Output device 220 may also be a display device capable of displaying still images or video, such as, without limitation, televisions, monitors, and projectors. Output device 220 can also be a device capable of outputting sound. Any device capable of playing or reading digital content in audio (e.g., music) or data (e.g., text or document) formats is also a possible output device 220.

[0090] A printer is frequently referred to herein as an example of an output device to simplify discussion or as the primary output device 220 in a particular implementation. However, it should be recognized that present invention applies also to other output devices 220 such as fax machines, digital copiers, display screens, monitors, televisions, projectors, voice output devices, among others.

[0091] Rendering content with an output device 220 refers to outputting the content on a specific output medium (e.g., papers, display screens etc). For example, rendering content with a printer generates an image on a substrate; rendering content with a display device generates an image on a screen; and rendering content with an audio output device generates sound.

[0092] A conventional printing system in general includes a raster image processor and a printer engine. A printer engine includes memory buffer, marking engine among other components. The raster image processor converts content into an image form suitable for printing; the memory buffer holds the rasterized image ready

for printing; and the marking engine transfers colorant to substrate (e.g., paper).

[0093] The raster image processor may be located within an output device (e.g. included in a printer controller 410) or externally implemented (in an information apparatus 200, external controller, servers etc). Raster image processor can be implemented as hardware, software, or a combination (not shown). As an example, raster image processor may be implemented in a software application or device driver in the information apparatus 200. Examples of raster image processing operations include image and graphics interpretation, rasterization, scaling, segmentation, color space transformation, image enhancement, color correction, halftoning, compression etc.

[0094] Fig. 4A illustrates a block diagram of one conventional printing system or printer 400A that includes a printer controller 410 and a printer engine 420A. The printer controller 410 includes an interpreter 402 and a raster image processor 406, and the printer engine 420 includes memory buffer 424A and a marking engine 426A.

[0095] Marking engine may use any of a variety of different technologies to transfer a rasterized image to paper or other media or, in other words, to transfer colorant to a substrate. The different marking or printing technologies that may be used include both impact and non-impact printing. Examples of impact printing may include dot matrix, teletype, daisywheel, etc. Non-impact printing technologies may include inkjet, laser, electrostatic, thermal, dye sublimation, etc.

[0096] The marking engine 426 and memory buffer 424 of a printer form its printer engine 420, which may also include additional circuitry and components, such as firmware, software or chips or chipsets for decoding and signal conversion, etc. Input to a printer engine 420 is usually a final rasterized printer-engine print data generated by a raster image processor 406. Such input is usually device dependent and printer or printer engine specific. The printer engine 420 may take this device dependent input and generate or render output pages (e.g. with ink on a substrate).

[0097] When a raster image processor is located inside an output device 220, it is usually included in a printer controller 410 (as shown in Fig. 4A). A printer controller 410 may interpret, rasterize, and convert input print data in the form of a page description language (e.g., PostScript, PCL), markup language

(e.g., XML, HTML) or other special document format or language (e.g. PDF, EMF) into printer-engine print data which is a final format, language or instruction that printer engine 420A can understand.

[0098] Print data sent to a printer with printer controller 410 is usually in a form (e.g. postscript) that requires further interpretation, processing or conversion. A printer controller 410 receives the print data, interprets, process, and converts the print data into a form that can be understood by the printer engine 420A. Regardless of the type of print data, conventionally, a user may need a device-specific driver in his or her information apparatus 200 in order to output the proper language, format, or file that can be accepted by a specific printer or output device 220.

[0099] Fig. 4B illustrates another conventional output device 400B. Output device 400B may be a printing device, a display device, a projection device, or a sound device. In the case that the output device is a printing device or a printer, the printer with reference to Fig. 4B does not include a printer controller 410. As an example, printer 400B may be a low-cost printer such as a desktop inkjet printer. RIP operations in this example may be implemented in a software application or in a device driver included in an information apparatus 200. The information apparatus 200 generates device dependent output data (or print data in case of a printer) by rasterizing and converting a digital document into output data (e.g. into a compressed CMKY data with one or more bits per pixel) that can be understood by an output engine (or printer engine in case of a printer) 420B.

[0100] Regardless of type or sophistication level, different output device 220 conventionally needs different printer drivers or output management applications in an information apparatus 200 to provide output capability. Some mobile devices with limited memory and processing power may have difficulty storing multiple device drivers or perform computational intensive RIP operations. It may also be infeasible to install a new device dependent or specific printer driver each time there is a need to print to a new printer. To overcome these difficulties, present invention provides several improvements to output device 220 or output system 250 as described in detail next.

[0101] In present invention, output device 220 may include an output controller 230 to help managing communication and negotiation processes with an information apparatus 200 and to

process output data. Output controller 230 may include dedicated hardware or software or combination of both for at least one output device 220. Output controller 230 may be internally installed, or externally connected to one or more output devices 220. The output controller 230 is sometimes referred to as a print server or output server.

[0102] Fig 5A and 5B illustrate two exemplary internal implementations of the output controller 230 of present invention. Fig 5A illustrates the implementation of an output controller 230 inside a conventional printer with reference to Fig. 4A, which includes a conventional printer controller 410(5A). The output controller 230(5A) includes an interpreter 510A component for decoding the intermediate output data of present invention; and a converter component 530A for converting one or more decoded output images into a printer-controller print data that is suitable for input to the printer controller 410(5A). An optional image processing component 520A may be included in the output controller 230(5A).

[0103] Fig 5B illustrates the implementation of an output controller 230 included internally in a conventional output device 220 with reference to Fig. 4B, which does not include a printer controller. The output controller 230(5B) includes an interpreter 510B component for decoding the intermediate output data of present invention; an image processor 520B component for performing one or more image processing operations such as color space conversion, color matching and digital halftoning; and an optional encoder 530B component to conform the processed output images into an output-engine output data that is suitable for input to the output engine 420B if the result of the image processing is not already in required form suitable for the output engine 420B.

[0104] In one implementation, output device 220 may include a communication unit 550 or adapter to interface with information apparatus 200. Output device 220 may sometimes include more than one communication unit 550 in order to support different interfaces, protocols, or communication standards with different devices. For example, output device 220 may communicate with a first information apparatus 200 through a Bluetooth interface while communicating with a second information apparatus 200 through a parallel interface. Examples of hardware components of a wireless communication unit are described in greater detail below with reference to Figs. 8A and 8B.

[0105] In one embodiment, output controller 230 does not include a communication unit, but rather utilizes or manages a communication unit residing in the associated output device 220 such as the illustration in Fig 5. In another embodiment, output controller 230 may include or provide a communication unit to output device 220 as shown in Fig. 6. For example, an output controller 230 with a wireless communication unit may be installed internally or connected externally to a legacy printer to provide it with wireless communication capability that was previously lacking.

[0106] Fig. 6 includes three functional block diagrams illustrating the hardware/software components of output controller 230 in three different implementations. Each components of an output controller 230 may include software, hardware, or combination. For example, an output controller 230 may include components using one or more or combinations of an application-specific integrated circuit (ASIC), a digital signal processor (DSP), a field programmable gate array (FPGA), firmware, system on a chip, and various communication chip sets. Output controller 230 may also contain embedded processors 670 A with software components or embedded application software to implement its feature sets and functionalities.

[0107] Output controller 230 may contain an embedded operating system 680. With an operating system, some or all functionalities and feature sets of the output controller 230 may be provided by application software managed by the operating system. Additional application software may be installed or upgraded to newer versions in order to, for example, provide additional functionalities or bug fixes. Fig. 6A and Fig 6C illustrates examples of implementation with an operating system 680 while Fig. 6B illustrates an example without the operating system 680 or the optional embedded processor 670.

[0108] Output controller 230 typically includes a memory unit 640, or may share a memory unit with, for example, printer controller 410. The memory unit and storage unit, such as ROM, RAM, flash memory and disk drive among others, may provide persistent or volatile storage. The memory unit or storage unit may store output device profiles, objects, codes, instructions or data (collectively referred to as software components) that implement the functionalities of the output controller 230. Part of the software components (e.g., output device profile) may be

uploaded to information apparatus 200 during or before a data output operation.

[0109] An output controller 230 may include a processor component 670A and 670C, a memory component 650, an optional storage component 640, and an optional operating system component 680. Fig. 6 shows one architecture or implementation where the memory 650, storage 640, processor 670, and operating system 680 components, if exist, can be share or accessed by other operational components in the output controller 230 such as the interpreter 610 and image processor 650. Fig. 6 shows two communication units 660A and 660B included in the output controller 230; however, the output controller 230 of present invention may include any number of communication units 660. It is also possible that the output controller does not contain any communication unit but rather utilizes the communication unit of an output device.

[0110] The output controller 230 may be connected externally to an output device 220 or integrated internally into the output device 220. Fig 5A and 5B illustrate implementations of output controller 230 inside an output device 220. The output controller 230, however, may also be implemented as an external box or station that is wired or wirelessly connected to an output device 220. An output controller 230 implemented as an external box or station to an output device 220 may contain its own user interface. One example of such an implementation is a print server connected to an output device 220 in an output system 250. Another configuration and implementation is to integrate or combine the functionalities of an output controller 230 with an existing printer controller 410 (referred to as "combined controller") if the output device 220 is a printer as shown with reference to Fig. 7C or 7F. A combined controller can also be internally integrated or externally connected to output device 220, and include functionalities of both printer controller 410 (e.g., input interpretation and or raster image processing) and output controller 230 of present invention. One advantage of this configuration is that the functionalities or components of output controller 230 and printer controller 410 may share the same resources, such as processing unit, memory unit, etc. Fig 6C illustrates an example of a combined controller implementation or output controller 230 where the printer controller 410C, interpreter 610C and converter 630C shares the use of the processor 670C, memory 650C and storage 640C, managed by an operating system 680C. Various exemplary implementations and configurations of an output controller 230 with respect to an

output device 220 or output system 250 are illustrated in further detail with reference to FIG 7.

[0111] Other possible implementations of output controller 230 may include, for example, a conventional personal computer (PC), a workstation, and an output server or print server. In these cases, the functionalities of output controller 230 may be implemented using application software installed in a computer (e.g., PC, server, or workstation), with the computer connected with a wired or wireless connection to an output device 220. Using a PC, server, workstation, or other computer to implement the feature sets of output controller 230 with application software is just another possible embodiment of the output controller 230 and in no way departs from the spirit, scope and process of the present invention.

[0112] The difference between output controller 230 and printer controller 410 should be noted. Printer controller 410 and output controller 230 are both controllers and are both dedicated hardware and or software for at least one output device 220. Output controller 230 refers to a controller with feature sets, capabilities, and functionalities of the present invention. A printer controller 410 may contain functions such as interpreting an input page description language, raster image processing, and queuing, among others. An output controller 230 may include part or all of the features of a printer controller 410 in addition to the feature sets, functionalities, capabilities, and processes of present invention.

[0113] Functionalities and components of output controller 230 for the purpose of providing universal data output may include or utilize:

- Components and operations to receive output data from a plurality of information apparatus 200; the output data may include an intermediate output data containing at least one rasterized image related to the data content intended for output.
- Components and operations to interpret and/or decode the intermediate output data.
- Components and operations to process the intermediate output data. Such components and operations may include image processing functions such as scaling, segmentation, color correction, color management, GCR, image enhancement, decompression, decryption, and or halftoning among others.

- Components and operations to generate an output-engine output data, the output-engine output data being in an output data format acceptable for input to an output engine.

- Components and operations to send the output-engine output data to the output engine.

[0114] When associated with an output device 220 that includes a printer controller 410, the output controller of present invention may further include or utilize:

- Components and operations to convert the intermediate output data into a printer-controller print data (e.g. a PDL such as PostScript and PCL), the printer-controller print data being in a format acceptable to a printer controller.

- Components and operations to send printer-controller print data to one or more printer controllers.

[0115] In addition to the above components and functionalities, output controller 230 may further include one or more of the following:

- Components and operations to communicate with one or more information apparatus 200 through a wired or wireless interface.

- Components and operations to communicate and or manage a communication unit included in the output controller 230 or output device 220.

- Components and operations to store at least part of an output device profile (a printer profile in case of a printer) in a memory component.

- Components and operations to respond to service request from an information apparatus 200 by transmitting at least part of an output device profile to the information apparatus requesting service. The output controller 230 may transmit the output device profiles or object in one or multiple sessions.

- Components and operations to broadcast or advertise the services provided by a host output device 220 to one or more information apparatus 200 that may request such services.

- Components and operations to implement payment processing and management functions by, for example, calculating and processing payments according to the services requested or rendered to a client (information apparatus 200).

- Components and operations to provide a user interface such as display screen, touch button, soft key, etc.

- Components and operations to implement job management functions such as queuing and spooling among others.

- Components and operations to implement security or authentication procedures. For example, the output controller 230 may store in its memory component (or shared memory component) an access control list, which specifies what device or user may obtain service from its host (or connected) output device 220. Therefore, an authorized information apparatus 200 may gain access after confirming with the control list.

[0116] When output controller 230 is implemented as firmware, or an embedded application, the configuration and management of the functionalities of output controller 230 may be optionally accomplished by, for example, using controller management software in a host computer. A host computer may be a desktop personal computer (PC), workstation, or server. The host computer may be connected locally or through a network to the output device 220 or the controller 230. Communication between the host computer and the output controller 230 can be accomplished through wired or wireless communication. The management application software in the host computer can manage the settings, configurations, and feature sets of the output controller 230. Furthermore, host computer's configuration application may download and or install application software, software components and or data to the output controller 230 for the purpose of upgrading, updating, and or modifying the features and capabilities of the output controller 230.

[0117] Output device 220 in one implementation includes or is connected to output controller 230 described above. Therefore, functionalities and feature sets provided by output controller 230 are automatically included in the functionalities of output device 220. The output device 220 may, however, implement or include other controllers and/or applications that provide at least partially the features and functionalities of the output controller 230.

[0118] Therefore, the output device 220 may include some or all of the following functionalities:

- Components and operations to receive multiple service requests or queries (e.g., a service request, a data query, an object or component query etc.) from a plurality of information apparatus 200 and properly respond to them by returning components, which may contain data, software, instructions and/or objects.

- Components and operations to receive output data from a plurality of information apparatus 200; the output data may include an intermediate output data containing one or more rasterized image related to the content intended for output.

- Components and operations to interpret and/or decoding the intermediate output data.

- Components and operations to process and/or convert the intermediate output data into a form (e.g. output-engine print data) suitable for rendering at an output engine associated with the output device.

- Components and operations to render a representation or an image related to the content onto an output medium (e.g. substrate or a display screen).

[0119] An output device 220 may further comprise optionally one or more of the following functionalities:

- Components and operations for establishing and managing a communication link with an information apparatus 200 requesting service; the communication link may include wired or wireless communication.

- Components and operations for storing at least part of an output device profile (e.g. printer profile) in a memory component.

- Components and operations to provide at least part of an output device profile (e.g., printer profile in case of a printer) to one or more information apparatus 200 requesting service. The output device 220 may transmit the output device profile in one or multiple sessions.

- Components and operations to advertise or broadcast services provided or available to one or more information apparatus 200.

- Components and operations to implement payment processing and management functions by, for example, calculating and processing payments according to the services requested by or rendered to a client (information apparatus 200).

- Components and operations to implement job management functionalities such as queuing and spooling among others.

- Components and operations to provide a user interface such as display screen touch button, soft key, power switch, etc.

- Components and operations to implement security or authentication procedures. For example, the output device 220 may store in its memory component (or a shared memory component) an access control list, which specifies what device or user may obtain service from it. Therefore, an authorized information apparatus 200 may gain access after confirming with the control list.

[0120] Figs. 7A - 7F illustrate various alternative configurations and implementations of output controller 230 with respect to an output device 230. Printer is sometimes used as an

exemplary output device 230 to demonstrate the various configurations. It should be understood, however, the output device 230 of present invention is not limited to printers.

[0121] As described with reference to Fig 4., a printer may or may not contain a printer controller 410. Printer 400A that includes a printer controller 410 typically has higher speed and is more expensive than printer 400B which does not include a printer controller 410.

[0122] Fig. 7A shows that output controller 230 may be cascaded externally to one or more printers (only one shown). Information apparatus 200 communicates with output controller 230A, which then communicates with output device 220 such as a printer 220A. The communication link between the output controller 230A and the printer 220A may be a wired link or a wireless link, as described above. Fig 6A and 6B illustrates two examples of functional component design of the output controller that can implement the configuration illustrated in Fig 7A. The Image processor 620 in this implementation is optional.

[0123] Fig. 7B shows another implementation in which output controller 230B is installed as one or more circuit boards or cards internally inside printer 220B. The output controller 230B may co-exist with printer controller 410 and other components of the printer 220B. One example of this implementation is to connect output controller 230B sequentially with the printer controller 310. Fig 5A shows as an example of an implementation.

[0124] Fig. 7C shows another implementation in which the functionalities of output controller 230 and printer controller 410 are combined into a single controller (referred to as "combined controller") 230C. In this implementation, it is possible to reduce the cost of material when compared to implementing two separate controllers as shown in Fig 7B. As an example, the combined controller 230C may share the same processors, memories, and storages to run the applications and functionalities of the two types of controllers and therefore, may have lower component costs when compared to providing two separate controllers. Fig. 6C illustrates an example of a combined controller functional component implementation.

[0125] Some printers do not include a raster image processor or printer controller 410, as illustrated in Fig. 4B. An example of this type of printer is a lower cost desktop inkjet printer. Input to an inkjet printer may consist of a compressed CMYK data (proprietary or published) with one or more bits per pixel input.

To output to a printer that does not include a printer controller, a device specific software application or a printer driver is typically required in an information apparatus 200 to perform raster image processing operations. Accordingly, output controller 230 can be implemented into a variety of output devices 220 and/or output systems 250 including printers that do not have printer controllers for performing raster image processing operations.

[0126] Fig. 7D and Fig 7E illustrate two implementations of output controller 230 in an output device 220 or system 250. The output device 230 or system 250 may include a display device, a projection device, an audio output device or a printing device. In the case when the output device 220D or 220E is a printer, it does not include a printer controller. Fig. 7D illustrates an implementation of an output controller 230D installed as an external component or "box" to output device 220D. For example, the output controller 230 may be implemented as an application in a print server or as a standalone box or station. In this configuration, some or all of raster image processing operations may be implemented in the output controller 230D. Output controller 230D receives intermediate output data from an information apparatus 200 and generates output-engine output data that is acceptable to the output engine included in the output device 220D. The output controller 230D may send the output data to the output device 220D through a wired or wireless communication link or connection. Fig 6A and 6B illustrates two example of functional component design of the output controller that can implement the configurations for both Fig. 7D and 7E.

[0127] Fig. 7E shows a fifth implementation of output controller 230E in which the output controller 230E is incorporated within output device 220E as one or more circuit boards or cards and may contain software and applications running on an embedded processor. As with output device 220D (Fig. 7D), output device 220E does not include a printer controller 410. Accordingly, the output controller 230E implements the functionalities and capabilities of present invention that may include part of or complete raster imaging processing operation.

[0128] Fig. 7F shows a sixth implementation, an external combined controller 230F that integrates the functionalities of a printer controller 310 and an output controller into a single external combined controller component or "box" 230F. The two controller functions may share a common processor as well as a common memory space to run applications of the two types of

controllers. Under this configuration, either information apparatus 200 or the combined controller 230F could perform or share at least part of raster image processing functionality. Fig. 6C shows an example of functional components of a combined controller 230F.

[0129] Another implementation of the combined controller 230F shown in Fig 7F is to use an external computing device (PC, workstation, or server) running one or more applications that include the functionality of output controller 230 and printer controller 410.

[0130] The above are examples of different implementations and configurations of output controller 230. Other implementations are also possible. For example, partial functionalities of output controller 230 may be implemented in an external box or station while the remaining functionalities may reside inside an output device 220 as a separate board or integrated with a printer controller 410. As another example, the functionalities of output controller 230 may be implemented into a plurality of external boxes or stations connected to the same output device 220. As a further example, the same output controller 230 may be connected to service a plurality of output devices 220

[0131] Figs. 8A and 8B are block diagrams illustrating two possible configurations of hardware/software components of wireless communication units. These wireless communication units can be implemented and included in information apparatus 200, in output controller 230 and in output device 220. Referring to Fig. 8A, a radio adapter 800 may be implemented to enable data/voice transmission among devices (e.g., information apparatus 200 and output device 220) through radio links. An RF transceiver 814 coupled with antenna 816 is used to receive and transmit radio frequency signals. The RF transceiver 814 also converts radio signals into and from electronic signals. The RF transceiver 814 is connected to an RF link controller 810 by an interface 812. The interface 812 may perform functions such as analog-to-digital conversion, digital-to-analog conversion, modulation, demodulation, compression, decompression, encoding, decoding, and other data or format conversion functions.

[0132] RF link controller 810 implements real-time lower layer (e.g., physical layer) protocol processing that enables the hosts (e.g., information apparatus 200, output controller 230, output device 220, etc.) to communicate over a radio link. Functions performed by the link controller 810 may include, without limitation, error detection/correction, power control, data packet

processing, data encryption/decryption and other data processing functions.

[0133] A variety of radio links may be utilized. A group of competing technologies operating in the 2.4 GHz unlicensed frequency band is of particular interest. This group currently includes Bluetooth, Home radio frequency (Home RF) and implementations based on IEEE 802.11 standard. Each of these technologies has a different set of protocols and they all provide solutions for wireless local area networks (LANs). Interference among these technologies could limit deployment of these protocols simultaneously. It is anticipated that new local area wireless technologies may emerge or that the existing ones may converge. Nevertheless, all these existing and future wireless technologies may be implemented in the present invention without limitation, and therefore, in no way depart from the scope of present invention.

[0134] Among the currently available wireless technologies, Bluetooth may be advantageous because it requires relatively lower power consumption and Bluetooth-enabled devices operate in piconets, in which several devices are connected in a point-to-multipoint system. Referring to Fig. 8B, one or more infrared (IR) adapters 820 may be implemented to enable data transmission among devices through infrared transmission. The IR adapters 820 may be conveniently implemented in accordance with the Infrared Data Association (IrDA) standards and specifications. In general, the IrDA standard is used to provide wireless connectivity technologies for devices that would normally use cables for connection. The IrDA standard is a point-to-point (vs. point-to-multipoint as in Bluetooth), narrow angle, ad-hoc data transmission standard.

[0135] Configuration of infrared adapters 820 may vary depending on the intended rate of data transfer. Fig. 8B illustrates one embodiment of infrared adapter 820. Transceiver 826 receives/emits IR signals and converts IR signals to/from electrical signals. A UART (universal asynchronous receiver/transmitter) 822 performs the function of serialization/deserialization, converting serial data stream to/from data bytes. The UART 822 is connected to the IR transceiver 826 by encoder/decoder (ENDEC) 824. This configuration is generally suitable for transferring data at relatively low rate. Other components (e.g., packet framer, phase-locked loop) may be needed for higher data transfer rates.

[0136] Figs. 8A and 8B illustrate exemplary hardware configurations of wireless communication units. Such hardware

components may be included in devices (e.g., information apparatus 200, output controller 230, output device 220, etc.) to support various wireless communications standards. Wired links, however, such as parallel interface, USB, Firewire interface, Ethernet and token ring networks may also be implemented in the present invention by using appropriate adapters and configurations.

[0137] FIG 9 is a logic flow diagram of an exemplary raster imaging process (RIP) 902 that can implement the universal output method of present invention. Content (e.g. digital document) 900 may be obtained and/or generated by an application running in an information apparatus 200. For example, a document browsing application may allow a user to download and or open digital document 900 stored locally or in a network node. As another example, a document creating or editing application may allow a user to create or edit digital documents in his/her information apparatus 200.

[0138] A client application 210 in the information apparatus may be in the form of a device driver, invoked by other applications residing in the information apparatus 200 to provide output service. Alternatively, the client application 210 of present invention may be an application that includes data output and management component, in addition of other functionalities such as content acquisitions, viewing, browsing, and or editing etc. For example, a client application 210 in an information apparatus 200 may itself include components and functions for a user to download, view and or edit digital document 900 in addition of the output management function described herein.

[0139] Raster image process method 902 allows an information apparatus 200 such as a mobile device to pervasively and conveniently output content (e.g. a digital document) to an output device 220 or system 250 that includes an output controller 230. A client application 210 in an information apparatus 200 may perform part of raster image processing operations (e.g. rasterization operation). Other operations of raster image processing such as halftoning can be completed by the output device 220 or by the output controller 230. In conventional data output methods, raster image processing is either implemented entirely in an information apparatus (e.g. a printer that does not include a printer controller with reference to Fig. 1A) or in an output device (e.g. a printer that includes a printer controller with reference to Fig. 1B). Present invention provides a more balanced approach where raster image process operations are shared between an information

apparatus 200 and an output device 220 or system 250. For example, content 600 may be processed (e.g. raster image processed) by different components or parts of an overall output system from a client application 210 to an output controller 230 before being sent to an output engine or a printer engine for final output in step 960. Because the raster image processing operations are not completely implemented in the information apparatus 200, there is less processing demand on the information apparatus 200. Therefore, present RIP process may enable additional mobile devices with less memory and processing capability to have data output capability.

[0140] In step 910, rasterization operation, a content (e.g. digital document), which may include text, graphics, and image objects, is conformed or rasterized to image form according to one or more rasterization parameters such as output size, bit depth, color space, resolution, number of color channels etc. During the rasterization operation, text and vector graphics information in the content are rasterized or converted into image or bitmap information according to a given set of rasterization parameters. Image information in the content or digital document may be scaled and or interpolated to fit a particular output size, resolution and bit depth etc. The rasterization parameters are in general device dependent, and therefore may vary according to different requirements and attributes of an output device 220 and its output engine. There are many ways to obtain device dependent rasterization parameters, as described in more detail below with reference to Fig 12A. Device dependent rasterization parameters, in one example, may be obtained from an output device profile stored in an information apparatus 200, an output device 220 or an output controller 230.

[0141] In an alternative implementation, rasterization parameters may be predetermined by a standard or specification. In this implementation, in step 910 the content 900 is rasterized to fit or match this predefined or standard rasterization parameters. Therefore, the rasterized output image becomes device independent. One advantage of being device independent is that the rasterized output image is acceptable with controllers, devices and/or output devices implemented or created with the knowledge of such standard or specification. A rasterized image with predefined or standardized attributes is usually more portable. For example, both the client application 210 and output device 220 or its output controller 230 may be preprogrammed to receive, interpret, and or

output raster images based on a predefined standard and/or specification.

[0142] Occasionally, a predefined standard or specification for rasterization parameters may require change or update. One possible implementation for providing an easy update or upgrade is to store information and related rasterization parameters in a file or a profile instead of hard coding these parameters into programs, components or applications. Client application 210, output controller 230, and/or the output device 220 can read a file or a profile to obtain information related to rasterization parameters. To upgrade or update the standard specification or defaults requires only replacing or editing the file or the profile instead of replacing a software application or component such as the client application 210.

[0143] In step 920 the rasterized content in image form is encoded into an intermediate output data. The intermediate output data, which describes the output content, may include image information, instructions, descriptions, and data (e.g. color profile). The rasterized output image may require further processing including one or more of compression, encoding, encryption, smoothing, image enhancement, segmentation, color correction among others before being stored into the intermediate output data. The output image in the intermediate output data may be encoded in any image format and with any compression technique such as JPEG, BMP, TIFF, JBIG etc. In one preferred embodiment, a mixed raster content (MRC) format and its related encoding and/or compression methods are used to generate the output image. The advantages of using MRC over other image formats and techniques may include, for example, better compression ratio, better data information retention, smaller file size, and or relatively better image quality among others.

[0144] In step 930, the intermediate output data is transmitted to the output device 220 or output system 250 for further processing and final output. The transmission of the intermediate output data may be accomplished through wireless or wired communication links between the information apparatus 200 and the output device 220 and can be accomplished through one or multiple sessions.

[0145] In step 940, the output device 220 or output system 250 receives the transmitted intermediate output data. The output device 220 or output system 250 may include an output controller 230 to assist communicating with the information apparatus 200

and/or processing the intermediate output data. Output controller 230 may have a variety of configurations and implementations with respect to output device 220 as shown in Fig. 7A-7F.

Interpretation process 940 may include one or more of parsing, decoding, decompression, decryption, image space conversion among other operations if the received intermediate output data requires such processing. An output image is decoded or retrieved from the intermediate output data and may be temporarily stored in a buffer or memory included in the output device/output system (220/250) or output controller 230 for further processing.

[0146] If the intermediate output data includes components with MRC format or encoding techniques, it may contain additional segmented information (e.g. foreground and background), which can be used to enhance image quality. For example, different techniques or algorithms in scaling, color correction, color matching, image enhancement, anti-aliasing and or digital halftoning among others may be applied to different segments or layers of the image information to improve output quality or maximize retention or recovery of image information. Multiple layers may later be combined or mapped into a single layer. These image processing and conversion components and/or operations can be included in the output controller 230 of present invention.

[0147] In step 950, the decoded or retrieved output image from the intermediate output data may require further processing or conversion. This may include one or more of scaling, segmentation, interpolation, color correction, GCR, black generation, color matching, color space transformation, anti-aliasing, image enhancement, image smoothing and or digital halftoning operations among others.

[0148] In an embodiment where the output device 220 does not include a printer controller, an output controller 230 or an output device 220 that includes output controller, after performing the remaining portion of RIP operations (e.g. color space conversion and halftoning) on the output image, may further convert the output data in step 950 into a form that is acceptable for input to a printer engine for rendering.

[0149] In an alternative embodiment where the output device 220 or the output system 250 includes a conventional printer controller, the output controller may simply decodes and or converts the intermediate output data (print data in this example) into format or language acceptable to the printer controller. For example, a printer controller may require as input a page

description language (e.g. PostScript, PCL, PDF, etc.), a markup language (HTML, XML etc) or other graphics or document format. In these cases, the output controller 230 may interpret, decompress and convert the intermediate print data into an output image that has optimal output resolution, bit depth, color space, and output size related to the printer controller input requirements. The output image is then encoded or embedded into a printer-controller print data (e.g. a page description language) and sent to the printer controller. A printer-controller print data is a print data that is acceptable or compatible for input to the printer controller. After the printer controller receives the printer-controller print data, the printer controller may further perform operations such as parsing, rasterization, scaling, color correction, image enhancement, halftoning etc on the output image and generate an appropriate printer-engine print data suitable for input to the printer engine.

[0150] In step 960, the output-engine output data or printer-engine print data generated by the output controller 230 or the printer controller in step 950 is sent to the output engine or printer engine of the output device for final output.

[0151] Fig. 10 illustrates a flow diagram of a universal data output process of the present invention that includes the raster image processing illustrated with reference to Fig. 9. A universal data output process allows an information apparatus 200 to pervasively output content or digital document to an output device. The data output process may include or utilize:

- A user interface component and operation where a user initiates an output process and provides an indication of the selected output content (e.g. digital document) for output.
- A client application component or operation that processes the content indicated for output, and generates an intermediate output data. The intermediate output data may include at least partly a raster output image description related to the content.
- An information apparatus component or operation that transmits the intermediate output data to one or more selected output device 220.
- An output device component (e.g. output controller) or operation that interprets the intermediate output data and may further process or convert the output data into a form more acceptable to an output engine for rendering of the content.

[0152] With reference to Fig. 10, a user in step 1000 may initiate the universal output method or process 1002. Typically, a

2025-10-20 10:50:00

user initiates the output process by invoking a client application 210 in his/her information apparatus 200. The client application 210 may be launched as an independent application or it may be launched from other applications 205 (such as from a document browsing, creating or editing application) or as part of or component of or a feature of another application 205 residing in the same information apparatus 200. When launched from another application 205, such as the case when the client application is a device driver or helper application, the client application 210 may obtain information, such as the content (e.g. digital document) from that other application 205. This can be accomplished, for example, by one or combinations of messages or facilitated through an operating system or a particular object or component model etc.

[0153] During output process 1002, a user may need to select one or more output devices 220 for output service. An optional discovery process step 1020 may be implemented to help the user select an output device 220. During the discovery process step 1020, a user's information apparatus 200 may (1) search for available output devices 220; (2) provide the user with a list of available output devices 220; and (3) provide means for the user to choose one or more output devices 220 to take the output job. An example of a discovery process 1020 is described below in greater detail with reference to Fig. 11.

[0154] The optional discovery process 1020 may sometimes be unnecessary. For example, a user may skip the discovery process 1020 if he or she already knows the output device (e.g., printer) 220 to which the output is to be directed. In this case, the user may simply connect the information apparatus 200 to that output device 220 by wired connections or directly point to that output device 220 in a close proximity such as in the case of infrared connectivity. As another example, a user may pre-select or set the output device or devices 220 that are used frequently as preferred defaults. As a result, the discovery process 1020 may be partially or completely skipped if the default output device 220 or printer is found to be available.

[0155] In stage 1030, the client application may interact with output device 220, the user, and/or other applications 205 residing in the same information apparatus 200 to (1) obtain necessary output device profile and/or user preferences, (2) perform functions or part of raster image processing operations such as rasterization, scaling and color correction, and/or (3) convert or encode at least partially the rasterized content (e.g. digital

document) into an intermediate output data. The processing and generation of the intermediate output data may reflect in part a relationship to an output device profile and/or user preferences obtained, if any. The intermediate output data generated by the client application 210 is then transmitted through wired or wireless local communication link(s) 240 to the output controller 230 included or associated with the selected output device 220 or output system 250. An exemplary client application process is described in greater detail with reference to FIG 12.

[0156] In step 1040, the output controller 230 of present invention receives the intermediate output data. In the case where the selected output device 230 does not include a printer controller, the output controller 230 of present invention may further perform processing functions such as parsing, interpreting, decompressing, decoding, color correction, image enhancement, GCR, black generation and halftoning among others. In addition, the output controller 230 may further convert or conform the intermediate output data into a form or format suitable for the output engine (e.g. printer engine in the case of a printer). The generated output-engine output data from the output controller is therefore, in general, device dependent and acceptable for final output with the output engine (or the printer engine in case of a printer) included in the selected output device 220 or output system 250.

[0157] In the case where the selected output device 220 is a printer, and when the printer includes or is connected to a printer controller, the output controller 230 may generate the proper language or input format required to interface with the printer controller (referred to as printer-controller print data). The printer controller may for example require a specific input such as a page description language (PDL), markup language, or a special image or graphics format. In these cases, the output controller 230 in step 1040 may interpret and decode the intermediate output data, and then convert the intermediate output data into the required printer-controller print data (e.g. PDL such as PostScript or PCL). The printer-controller print data generated by the output controller is then sent to the printer controller for further processing. The printer controller may perform interpretation and raster image processing operations among other operations. After processing, the printer controller generates a printer-engine print data suitable for rendering at the printer engine.

[0158] In either case, the output controller 230 or printer controller generates an output-engine output data that is suitable for sending to or interfacing with the output engine or the printer engine included in the output device for rendering. The output data may be temporarily buffered in components of the output device 220. An implementation of the output device process 1040 is described in greater detail with reference to FIG 13.

[0159] The steps included in the universal pervasive output process 1002 may proceed automatically when a user requests output service. Alternatively, a user may be provided with options to proceed, cancel, or input information at each and every step. For example, a user may cancel the output service at any time by, for example, indicating a cancellation signal or command or by terminating the client application 210 or by shutting down the information apparatus 200 etc.

[0160] Fig. 11 is a flow diagram of an example of a discovery process 720, which may be an optional step to help a user locate one or more output devices 220 for an output job. The discovery process 1020 may, however, be skipped partially or entirely. Implementation of discovery process 1020 may require compatible hardware and software components residing in both the information apparatus 200 and the output device 220. The information apparatus 200 may utilize the client application 210 or other application 205 in this process. The discovery process 1020 may include:

- An information apparatus 200 communicating with available output devices 220 to obtain information and attributes relating to the output device 220 and or its services such as output device capability, feature sets, service availability, quality of service, condition.

- An Information apparatus 200 provides the user information on each available and or compatible output devices 220.

- A user selects or the client application 210 (automatically or not) selects one or more output devices 220 for the output service from the available or compatible output devices 220.

[0161] Various protocols and or standards may be used during discovery process 1020. Wireless communication protocols are preferred. Wired communication, on the other hand, may also be implemented. Examples of applicable protocols or standards may include, without limitation, Bluetooth, HAVi, Jini, Salutation, Service Location Protocol, and Universal Plug-and-play among others. Both standard and proprietary protocols or combination may be implemented in the discovery process 1020. However, these

different protocols, standards, or combination shall not depart from the spirit and scope of present invention.

[0162] In one implementation an application (referred here for simplicity of discussion as a "communication manager," not shown) residing in the information apparatus 200 helps communicate with output device 220 and manages service requests and the discovery process 1020. The communication manager may be a part of or a feature of the client application 210. Alternatively or in combination, the communication manager may also be a separate application. When the communication manager is a separate application, the client application 210 may have the ability to communicate, manage or access functionalities of the communication manager.

[0163] The discovery process 1020 may be initiated manually by a user or automatically by a communication manager when the user requests an output service with information apparatus 200.

[0164] In the optional step 1100, a user may specify searching or matching criteria. For example, a user may indicate to search for color printers and or printers that provide free service. The user may manually specify such criteria each time for the discovery process 1020. Alternatively or in combination, a user may set default preferences that can be applied to a plurality of discovery processes 1020. Sometimes, however, no searching criteria are required: the information apparatus 200 may simply search for all available output devices 220 that can provide output service.

[0165] In step 1101, information apparatus 200 searches for available output devices 220. The searching process may be implemented by, for example, an information apparatus 200 (e.g. with the assistance of a communication manager) multi-casting or broadcasting or advertising its service requests and waiting for available output devices 220 to respond. Alternatively or in combination, an information apparatus 200 may "listen to" service broadcasts from one or more output devices 220 and then identify the one or more output devices 220 that are needed or acceptable. It is also possible that multiple output devices 220 of the same network (e.g., LAN) register their services with a control point (not shown). A control point is a computing system (e.g., a server) that maintains records on all service devices within the same network. An information apparatus 200 may contact the control point and search or query for the needed services.

[0166] In step 1102, if no available output device 220 is found, the communication manager or the client application 210 may

provide the user with alternatives 1104. Such alternatives may include, for example, aborting the discovery process 1020, trying discovery process 1020 again, temporarily halting the discovery process 1020, or being notified when an available output device 220 is found. As an example, the discovery process 1020 may not detect any available output device 220 in the current wired/wireless network. The specified searching criteria (if any) are then saved or registered in the communication manager. When the user enters a new network having available output devices 220, or when new compatible output devices 220 are added to the current network, or when an output device 220 becomes available for any reason, the communication manager may notify the user of such availability.

[0167] In step 1106, if available output devices 220 are discovered, the communication manager may obtain some basic information, or part of or the entire output device profile, from each discovered output device 220. Examples of such information may include, but not limited to, device identity, service charge, subscription, service feature, device capability, operating instructions, etc. Such information is preferably provided to the user through the user interface (e.g., display screen, speaker, etc.) of information apparatus 200.

[0168] In step 1108, the user may select one or more output devices 220 based on information provided, if any, to take the output job. If the user is not satisfied with any of the available output device 220, the user may decline the service. In this case, the user may be provided with alternatives such as to try again in step 1110 with some changes made to the searching criteria. The user may choose to terminate the service request at any time. In step 1112, with one or more output devices 220 selected or determined, the communication link between information apparatus 200 and the selected output device or devices 220 may be "locked". Other output devices 220 that are not selected may be dropped. The output process 1020 may then proceed to the client application process of step 1030 of Fig. 10.

[0169] Fig. 12A is a flow diagram of an exemplary client application process with reference to step 1030 of Fig. 10. A client application process 1202 for universal output may include or utilize:

- A client application 210 that obtains content (e.g. digital document) intended for output.
- A client application 210 that obtains output device parameters (e.g. rasterization parameters, output job parameters).

One example of implementation is to obtain the output device parameters from an output device profile (e.g. printer profile), which includes device dependent parameters. Such profile may be stored in an output controller 230, output device 220 or information apparatus 200.

- A client application 210 that may optionally obtain user preferences through (1) user's input (automatic or manual) or selections or (2) based on preset preference or pre-defined defaults or (3) combination of the above.

- A client application 210 that rasterizes at least part of the content intended for output (e.g. a digital document) according to one or more rasterization parameters obtained from previous steps such as through output device profile, user selection, predefined user preferences, predefined default or standard etc.

- A client application 210 that generates an intermediate output data containing at least part of the rasterized image related at least partly to the content intended for output.

- A client application that transmits the intermediate output data to an output device 220 or output controller 230 for further processing and or final output.

[0170] A client application 210 may obtain content (e.g. digital document) 900 or a pointer or reference to the content in many ways. In a preferred embodiment, the client application 210 is in the form of a device driver or an independent application, and the content or its reference can be obtained by the client application 210 from other applications 205 in the same information apparatus 200. To illustrate an example, a user may first view or download or create a digital document by using a document browsing, viewing and or editing application 205 in his/her information apparatus 200, and then request output service by launching the client application 210 as a device driver or helper application. The client application 210 communicates with the document browsing or editing application to obtain the digital document or reference to the digital document. As another example, the client application 210 is an independent application and it launches another application to help locate and obtain the digital document for output. In this case, a user may first launch the client application 210, and then invoke another application 205 (e.g. document editing and or browsing application) residing in the same information apparatus 200 to view or download a digital document. The client application 210 then communicates with the document

browsing or editing application to obtain the digital document for output.

[0171] In another embodiment, the client application 210 itself provides multiple functionalities or feature sets including the ability for a user to select the content (e.g. digital document) for output. For example, the client application 210 of present invention may provide a GUI where a user can directly input or select the reference or path of a digital document that the user wants to output.

[0172] In order to perform rasterization operation on content (e.g. digital document) 900, the client application 210 in step 1210 needs to obtain device dependent parameters of an output device 220 such as the rasterization parameters. Device dependent parameters may be included in an output device profile. A client application 210 may obtain an output device profile or rasterization parameters in various ways. As an example, an output device profile or rasterization parameters can be obtained with one or combination of the following:

- The client application communicates with an output device 220 to upload output device profile or information related to one or more rasterization parameters.
- The client application 210 obtains the output device profile from a network node (e.g. server).
- A user selects an output device profile stored in the user's information apparatus 200.
- The client application 210 automatically retrieves or uses a default profile, predefined standard values or default values among others.
- The client application 210 obtains output device parameters by calculating, which may include approximation, based at least partly on the information it has obtained from one or combination of an output device 220, a user, default values, and a network node.

[0173] It is important to note that step 1210 is an optional step. In some instance, part of or the entire output device profile or related device dependent information may have been already obtained by the client application 210 during the prior optional discovery process (step 1020 in FIG 10). In this case, step 1210 may be partially or entirely skipped.

[0174] In one implementation, the client application 210 communicates with one or more output devices 220 to upload output device profiles stored in the memory or storage components of those

one or more output devices 220 or their associated one or more output controllers 230. In some instance, the uploaded output device profile may contain partially or entirely references or pointers to device parameters instead of the device parameters themselves. The actual output device parameters may be stored in a network node or in the information apparatus 200, where they can be retrieved by the client application 210 or by other applications 205 using the references or pointers. It should be noted that a plurality of information apparatuses 200 may request to obtain output device profile or profiles from the same output device 220 at the same time or at least during overlapping periods. The output device 220 or its associated output controller 230 may have components or systems to manage multiple communication links and provide the output device profile or profiles concurrently or in an alternating manner to multiple information apparatuses 200. Alternatively, an output device 220 may provide components or systems to queue the requests from different information apparatuses 200 and serve them in a sequential fashion according to a scheme such as first come first served, quality of service, etc. Multi-user communication and service management capability with or without queuing or spooling functions may be implemented by, for example, the output controller 230 as optional feature sets.

[0175] In another implementation, one or more output device profiles may be stored locally in the information apparatus 200. The client application 210 may provide a GUI where a user can select a profile from a list of pre-stored profiles. As an example, the GUI may provide the user with a list of output device names (e.g. makes and models), each corresponding to an output device profile stored locally. When the user selects an output device 220, the client application 210 can then retrieve the output device profile corresponding to the name selected by the user.

[0176] In certain cases, during a discovery or communication process described earlier, the client application 210 may have already obtained the output device ID, name, or reference or other information in a variety of ways described previously. In this case, the client application 210 may automatically activate or retrieve an output device profile stored in the information apparatus 200 based on the output device ID, name, or reference obtained without user intervention.

[0177] In yet another implementation, the client application 210 may use a set of pre-defined default values stored locally in a user's information apparatus 200. Such defaults can be stored in

one or more files or tables. The client application 210 may access a file or table to obtain these default values. The client application 210 may also create or calculate certain default values based on the information it has obtained during previous steps (e.g. in optional discovery process, based on partial or incomplete printer profile information obtained, etc). A user may or may not have an opportunity to change or overwrite some or all defaults.

[0178] Finally, if, for any reason, no device dependent information is available, the client application 210 may use standard output and rasterization parameters or pre-defined default parameters. The above illustrates many examples and variations of implementation, these and other possible variations in implementation do not depart from the scope of the present invention.

[0179] In step 1220, the client application 210 may optionally obtain user preferences. In one exemplary implementation, the client application 210 may obtain user preferences with a GUI (graphical user interface). For simplicity, a standard GUI form can be presented to the user independent of the make and model of the output device 220 involved in the output process. Through such an interface, the user may specify some device independent output parameters such as page range, number of cards per page, number of copies, etc. Alternatively or in combination, the client application 210 may also incorporate output device-dependent features and preferences into the GUI presented to the user. The device-dependent portion of the GUI may be supported partly or entirely by information contained in the output device profile obtained through components and processes described in previous steps. To illustrate, device dependent features and capabilities may include print quality, color or grayscale, duplex or single sided, output page size among others.

[0180] It is preferred that some or all components, attributes or fields of user preferences have default values. Part or all default values may be hard-coded in software program in client application 210 or in hardware components. Alternatively, the client application 210 may also access a file to obtain default values, or it may calculate certain default values based on the information it has obtained during previous steps or components (e.g. from an output device profile). A user may or may not have the ability to pre-configure, or change or overwrite some or all defaults. The client application 210 may obtain and use some or all defaults with or without user intervention or knowledge.

[0181] In step 1230, the client application 210 of present invention performs rasterization operation to conform a content (e.g. a digital document), which may includes objects and information in vector graphics, text, and images, into one or more output images in accordance with the rasterization parameters obtained in previous steps. During rasterization process, text and vector graphics object or information in the content is rasterized or converted into image or bitmap form according to the given set of rasterization parameters. Image information in the content may require scaling and interpolation operations to conform the rasterization parameters. Rasterization process may further include operations such as scaling, interpolation, segmentation, image transformation, image encoding, color space transformation etc. to fit or conform the one or more output images to the given set of rasterization parameters such as target output size, resolution, bit depth, color space and image format etc.

[0182] In step 1240, the client application 210 generates an intermediate output data that includes the rasterized one or more output images. The intermediate output data of the present invention may contain image information, instructions, descriptions, and data such as color profile among others. Creating and generating intermediate output data may further include operations such as compression, encoding, encryption, smoothing, segmentation, scaling and or color correction, among others. The image or images contained in an intermediate output data may be variously encoded and/or implemented with different image formats and/or compression methods (e.g. JPEG, BMP, TIFF, JBIG etc or combination). One preferred implementation is to generate or encode the output image in the intermediate output data with mixed raster content (MRC) description. The use of MRC in the data output process of present invention provides opportunities to improve the compression ratio by applying different compression techniques to segmented elements in the content. In addition, MRC provides opportunities to maintain more original content information during the encoding process of the output image and, therefore, potentially improve output quality.

[0183] In step 1250, the client application 210 transmits intermediate output data to an output device 220 through local communication link 240. The communication link may be implemented with wired or wireless technologies and the transmission may include one or multiple sessions.

[0184] It should be recognized that FIG 12A illustrates one example of a client application process 1030 in the data output method 1002 of present invention. Other implementations with more or less steps are possible, and several additional optional processes not shown in Fig 12 may also be included in the client application process 1030. Use of these different variations, however, does not result in a departure from the scope of the present invention. As an example, an optional authentication step may be included when the selected output device 220 provides service to a restricted group of users. Various authentication procedures may be added in step 1210 when client application 210 obtains output device profile by communicating with an output device or an output controller. As another example, authentication procedures may also be implemented in step 1250 when the client application transmits intermediate output data to one or more output devices 220 or output controllers 230. A simple authentication may be implemented by, for example, comparing the identity of an information apparatus 200 with an approved control list of identities stored in the output device 220 or output controller 230. Other more complex authentication and encryption schemes may also be used. Information such as user name, password, ID number, signatures, security keys (physical or digital), biometrics, fingerprints, voice among others, may be used separately or in combination as authentication means. Such identification and or authentication information may be manually provided by user or automatically detected by the selected output device or devices 220 or output controller 230. With successful authentication, a user may gain access to all or part of the services provided by the output device 220. The output device profile that the client application 210 obtains may vary according to the type or quality of service requested or determined. If authentication fails, it is possible that a user may be denied partially or completely access to the service. In this case, the user may be provided with alternatives such as selecting another output device 220 or alternative services.

[0185] Another optional process is that a user may be asked to provide payment or deposit or escrow before, during or after output service such as step 1210 or 1250 with reference to Fig. 12. Examples of payment or deposit may include cash, credit card, bankcard, charge card, smart card, electronic cash, among others. The output controller 220 may provide payment calculation or

transaction processing as optional feature sets of present invention.

[0186] FIG 12B illustrates another exemplary client application output process 1030 with which an information apparatus 200 can pervasively and universally output content to one or more output devices 220 associated with or equipped with an output controller 230 of present invention.

[0187] The process illustrated in FIG 12B is similar to the process described in FIG 12A except that step 1210, obtaining output device profile, is skipped. In this embodiment, the client application 210 utilizes a set of hard-coded, standard or pre-defined output device parameters including rasterization parameters with which the client application 210 can perform rasterization operation and other required image processing functions. Users may be provided with the option of changing these parameters or inputting alternative parameters. Rasterization parameters include output size, output resolution, bit depth, color space, color channels, scale factors etc. These pre-defined parameters typically comply with a specification or a standard. The same specification and standard may also defined or describe at least partly the intermediate output data. Predefined standard parameters can be stored in a file or profile in an information apparatus 200, an output controller 230, and/or in an output device 220 for easy update or upgrade.

[0188] In client output process 1204, since the rasterization parameters are predefined, the client application 210 may not need to upload printer profiles from the selected output device 230. Consequently, no two-way communication between the information apparatus 200 and the output device or devices 220 is necessary in this process 1204 when compared with process 1202 illustrated in Fig. 12A. The client application 210 performs rasterization operation 1225 based on standard and/or predefined parameters and generates a rasterized output image with predefined or standard properties of those rasterization parameters. The resulting intermediate output data, which includes at least one rasterized output image, is transmitted from the information apparatus 200 to an output device 220 in step 1250 or to its associated output controller 230 for rendering or output. The intermediate output data generated in process 1202 in general is less device dependent compared to the intermediate output data generated in the process 1202 shown in Fig 12A. The output controller 230 included or associated with the output device 220 may be preprogrammed to

interpret the raster output image, which includes properties or attributes that correspond to those standard or predefined parameters.

[0189] The standard or predefined rasterization parameters may be hard coded or programmed into the client application 210 and/or the output controller 230. However, instead of hard coding those parameters, one technique to facilitate updates or changes is to store those standard parameters in a default file or profile. The standard or predefined parameters contained in the file or profile can be retrieved and utilized by applications in an information apparatus 200 (e.g. client application 210) and/or by applications or components in an output device 220 or the output controller 230. In this way, any necessary updates, upgrades or required changes to those predefined or standard parameters can be easily accomplished by replacing or modifying the file or profile instead of modifying or updating the program, application or components in the information apparatus 200, output device 220 and/or output controller 230.

[0190] A client application process 1204 providing universal output capability to information apparatus 200 may include or utilize:

- A client application 210 that obtains content (e.g. digital document) intended for output.

- A client application 210 that optionally obtains user preferences (in step 1220) through (1) user's input (automatic or manual) or selections or (2) based on preset preference or pre-defined defaults or (3) combination of the above.

- A client application 210 that rasterizes content (in step 1230 or 1225) according to pre-defined or standard rasterization parameters.

- A client application 210 that generates intermediate output data (in step 1240) for rendering or output at an output device 220; the intermediate output data containing at least partially a rasterized image related to the content intended for output.

- A client application 210 that transmits the intermediate output data to an output device 220 (in step 1250) for further processing and final output.

[0191] One advantage of the client output process 1204 of Fig 12B compared to the process 1202 illustrated in Fig 12A is that the generated intermediate output data is in general less device dependent. The device independent attribute allows the intermediate output data to be more portable and acceptable to more output

devices equipped or associated with output controllers. Both data output processes (1202 and 1204) enable universal output; allowing a user to install a single client application 210 or components in an information apparatus 200 to provide output capability to more than one output device 220.

[0192] FIG 13A illustrates one example of an output device process 1302 and its associated raster imaging method of present invention. In this output device process 1302, an output device 220 is capable of receiving an intermediate output data from an information apparatus 200. The output device process 1302 and its operations may include or utilize:

- An output device/system or output controller that receives intermediate output data (in step 1300). The intermediate output data includes at least partially a raster output image describing at least part of the content for rendering at the output device 220 or system 250.

- An output device/system or output controller that interprets (in step 1310) the intermediate output data; in one preferred embodiment, the intermediate output data includes an output image utilizing one or more MRC formats or components.

- An output device/system or output controller that performs image processing operation (in step 1320) on the raster image. The image processing operation may include but not limited to image decompression, scaling, halftoning, color matching, among others.

- An output device/system or output controller that converts and or generates (in step 1330) output-engine output data that is in a format or description suitable for input to an output engine (e.g. printer engine in case of a printer) included in an output device 220.

- An output engine in an output device 220 that renders or generates a final output (e.g. the output-engine output data) in step 1370.

[0193] The output device 220 or output system 250 may include an output controller 230 internally or externally to assist the management and operation of the output process 1302. As shown in Fig 7, there are many possible configurations and implementations of an output controller 230 associated to an output device 220. Herein and after, output controller 230 is regarded as an integral part of the output device to which it is attached. Hence, the following described output device operations may be partially or completely performed by the output controller associated with it.

[0194] In step 1300, output device process 1302 is initiated by client application 210 transmitting an intermediate output data to output device 220 or output system 250. In step 1310, the output device 220 reads and interprets the intermediate output data, containing at least one raster output image relating to the content intended for output. During the reading and interpretation process 1310, the output device 220 may include components that parse the intermediate output data and perform operations such as decompression, decoding, and decryption among others. The output image may be variously encoded and may include one or more compression methods.

[0195] In the event that the method of image encoding includes MRC format, then, in one example implementation, during decoding and mapping of the output image in step 1310, the lower resolution layer and information in an image that includes MRC may be mapped, scaled or interpolated to a higher-resolution output image to produce a better image quality. Therefore, step 1310, in the event that the intermediate output data includes MRC component, each layer in an MRC image can be decompressed, processed, mapped and combined into a single combined output image layer. Step 1310 may also include scaling, color space transformation, and/or interpolation among others. In addition to the possibility of mapping methods using different scaling and interpolation ratio with different layers, another advantage of using MRC is that segmentation information contained in MRC can be utilized to apply different image processing and enhancement techniques to data in different layers of an MRC image in step 1320.

[0196] In step 1320, the output device 220 may further perform image processing operations on the decoded output image. These image processing operations may include, for example, color correction, color matching, image segmentation, image enhancement, anti-aliasing, image smoothing, digital watermarking, scaling, interpolation, and halftoning among others. The image processing operations 1320 may be combined or operated concurrently with step 1310. For example, while each row, pixel, or portion of the image is being decoded and or decompressed, image processing operations 1320 is applied. In another implementation, the image processing 1320 may occur after the entire output image or a large portion of the image has been decoded or decompressed.

[0197] If the intermediate output data includes MRC component, then in step 1320, there are additional opportunities to improve image quality. An image encoded in MRC contains segmented

information that a traditional single layer image format does not usually have. As an example, foreground can be in one layer, and background in another. As another example, chrominance information may be in one layer and luminance may be in another. This segmented information in MRC may be used to apply different or selective image processing methods and algorithms to different layers or segments to enhance image quality or retain or recover image information. Different image processing techniques or algorithms may include color matching, color correction, black generation, halftoning, scaling, interpolation, anti-aliasing, smoothing, digital watermarking etc. For example, one can apply colorimetric color matching to foreground information and perceptual color matching to background information or vice versa. As another example, error diffusion halftoning can be applied to foreground and stochastic halftoning can be applied to background or vice versa. As yet another example, bi-cubic interpolation can be applied to a layer and bi-linear or minimum distance interpolation can be applied to a different layer.

[0198] In step 1330, the output device 220 or the output controller 230 may convert the processed image (e.g. halftoned) into a form acceptable to the output engine of output device 220. This conversion step is optional, depending on the type, format and input requirement of a particular output device engine (e.g. printer engine in case of a printer). Different output engines may have different input raster image input requirements. As an example different output engines may require different input image formats, number of bits or bytes per pixel, compression or uncompressed form, or different color spaces (e.g. such as RGB, CMY, CMYK, or any combination of Hi-Fi color such as green, orange, purple, red etc). Incoming raster image data can be encoded in a row, in a column, in multiple rows, in multiple columns, in a chunk, in a segment, or a combination at a time for sending the raster data to the output engine. In some cases, step 1330 may be skipped if the result of step 1320 is already in a form acceptable to the output device engine. In other cases, however, further conversion and or processing may be required to satisfy the specific input requirement of a particular output device engine.

[0199] It is important to note that the above described processing from step 1310 to step 1330 may require one or more memory buffers to temporarily store processed results. The memory buffer can store or hold a row, a column, a portion, or a chunk, of the output image in any of the steps described above. Storing and

retrieving information into and from the memory buffer may be done sequentially, in an alternating fashion, or in an interlaced or interleaved fashion among other possible combinations. Step 1310 to step 1330 operations can be partially or completely implemented with the output controller 230.

[0200] In step 1370, the output device engine included in the output device 220 or output system 250 receives the output-engine output data generated in step 1330 or step 1320. The output-engine output data is in a form that satisfies the input requirements and attributes of the output engine, such as color space, color channel, bit depth, output size, resolution, etc. The output engine then takes this output-engine output data and outputs or renders the data content through its marking engine or display engine.

[0201] One advantage of data output method 1002 that includes output device process 1302 is that it has less processing requirements on an information apparatus 200 compared to conventional process with reference to Fig 1A, and therefore, enables more information apparatus 200 with relatively lower processing power and memory space to have output capability.

[0202] For example, some image processing functions, such as halftoning (e.g. error diffusion) may require substantial processing and computing power. In data output process 1002 that includes output device process 1302, halftoning is performed in step 1320 by an output device component (e.g. the output controller 230) included in the output device 220 or the output system 250, not in the information apparatus 200; therefore reducing the computational requirements for the information apparatus 200. Another advantage of data output 1302 is that the intermediate output data is less device dependent than the output data generated by conventional output method 102 with reference to Fig 1A. The device independence provides opportunity to allow a single driver or application in an information apparatus 200 to output intermediate output data to a plurality of output devices 220 that include output controllers 230.

[0203] Some output devices 220 may contain a printer controller 410. An example of this type of output device or printer is a PostScript printer or PCL printer among others. FIG 13B illustrates an example of an output device process 1304 with a printer that includes a printer controller 410. As discussed in Fig 1, a printer with a printer controller requires input such as page description language (e.g. PostScript, PCL etc.), markup language

(HTML, XML etc), special image format, special graphics format, or a combination, depending on the type of the printer controller.

[0204] There are many printing system configurations for providing the data output capability and process to a printer or a printing system that includes a printer controller. In one example, the existing printer controller in the output device 220 may incorporate the feature sets provided by the output controller to form a "combined controller" as described previously with reference to Figs 7C and 7F. In another example, the output controller 230 of present invention may be connected sequentially or cascaded to an existing printer controller; the output controller 230 can be internally installed (with reference to FIG. 7B) or externally connected (with reference to FIG. 7A) to the output device 220. For output device 220 that includes a printer controller, the output controller 230 may simply decode the intermediate output data in step 1310 and then convert it into a form acceptable for input to the printer controller in step 1350.

[0205] An output device process 1304 and operations for an output device 220 or system 250 that includes a printer controller 410 may include or utilize:

- An output controller 230 or components in an output device 220 or system 250 that receives an intermediate print data or output data (with reference to step 1300), the intermediate print data includes at least a raster image related at least in part to the content for rendering at the output device 220.
- An output controller 230 or components in an output device 220 or system 250 that interprets the intermediate output data (with reference to step 1310); in one preferred embodiment, the intermediate output data includes an output image utilizing one or more MRC format or components.
- An output controller 230 or components in an output device 220 or system 250 that converts the intermediate output data into a printer-controller print data (with reference to step 1350); the printer-controller print data includes a format or language (e.g. PDL, PDF, HTML, XML etc.) that is acceptable or compatible to the input requirement of a printer controller.
- A printer controller or components in an output device 220 or system 250 that receives a printer controller print data; the printer controller may parse, interpret and further process (e.g. rasterization, scaling, image enhancement, color correction, color matching, halftoning etc.) and convert the printer-controller print data into a printer-engine print data (with reference to step

1360); the printer-engine print data comprising of a format or description acceptable for input to a printer engine in the output device 220 or the output system 250.

- A printer engine or components in an output device 220 or system 250 that renders or generates a final output (with reference to step 1370) with the input printer engine print data.

[0206] In output device process 1304, step 1300 (receiving intermediate output data) and step 1310 (interpret intermediate output data) are identical to step 1300 and step 1310 in output device process 1302, which have been described in previous sections with reference to Fig. 13A.

[0207] In step 1350, the output controller 230 converts the intermediate print data into a printer-controller print data that is in a form compatible or acceptable for input to a printer controller. For example, a printer controller may require as input a specific page description language (PDL) such as PostScript. The output controller 230 then creates a PostScript file and embeds the output image generated or retrieved in step 1310 into the PostScript file. The output controller 230 can also create and embed the output image from step 1310 into other printer controller print data formats, instructions or languages.

[0208] In step 1360, the printer controller receives printer-controller print data generated in step 1350 that includes an acceptable input language or format to the printer controller. The printer controller may parse, interpret, and decode the input printer-controller print data. The printer controller may further perform raster image processing operations such as rasterization, color correction, black generation, GCR, anti-aliasing, scaling, image enhancement, and halftoning among others on the output image. The printer controller may then generate a printer-engine print data that is suitable for input to the printer engine. The type and or format of printer-engine print data may vary according to the requirement of a particular printer engine.

[0209] It is important to note that the above described process from step 1310 to step 1360 may require one or more memory buffer to temporarily store processed results. The memory buffer can store or hold a row, a column, a portion, or a chunk, of the output image in any of the steps described above. Storing and retrieving information into and from the memory buffer may be done sequentially, alternated, or in an interlaced or interleaved fashion among other possible combinations. Process and operations

1054549-011802

of step 1310 to step 1360 can be implemented with output controller 230.

[0210] In step 1370, the printer engine included in the output device 220 or output system 250 generates or renders the final output based on the printer-engine print data generated in step 1360. For example, the printer-engine print data may be in CMY, CMYK, and RGB etc, and this may be in one or more bits per pixel format, satisfying the size and resolution requirement of the printer engine. The printer engine included the output device 220 may take this print data and generate or render an output page through its marking engine.

[0211] Having described and illustrated the principles of our invention with reference to an illustrated embodiment, it will be recognized that the illustrated embodiment can be modified in arrangement and detail without departing from such principles. In view of the many possible embodiments to which the principles of our invention may be applied, it should be recognized that the detailed embodiments are illustrative only and should not be taken as limiting the scope of our invention. Rather, I claim as my invention all such embodiments as may come within the scope of the following claims and equivalents thereto.

[0212] Unless the context indicates otherwise, a reference in a claim to the number of instances of an element, be it a reference to one instance or more than one instance, requires at least the stated number of instances of the element but is not intended to exclude from the scope of the claim a structure or method having more instances of that element than stated. Specifically, but without limitation, a reference in a claim to an or one output device or system, to an or one image, or to a or one rasterization parameter is not intended to exclude from the scope of the claim a structure or method having, including, employing or supplying two or more output devices or system, images or rasterization parameters.